

## МОДУЛЬ РАСЧЕТА ОПТИМАЛЬНОГО МАРШРУТА ДВИЖЕНИЯ НА ОБЩЕСТВЕННОМ ТРАНСПОРТЕ

### Аннотация

*Настоящая статья посвящена программной реализации задачи поиска оптимального маршрута движения на общественном транспорте из пункта «А» в пункт «Б». Разработанная программа предназначена для использования в составе комплексной информационной системы информирования о маршрутах общественного транспорта. По заданным перечням начальных и конечных точек (в качестве начальных точек выступают терминалы, устанавливаемые на остановочных комплексах и подключенные к информационной системе; в качестве конечных – остановочные комплексы или достопримечательности города) и данным о движении общественного транспорта программа рассчитывает оптимальный маршрут между всеми парами начальных и конечных точек. Оптимальные маршруты средствами комплексной информационной системы отображаются на терминалах остановочных комплексов по запросам жителей и гостей города. В статье рассматриваются алгоритм и реализованное решение задачи поиска кратчайшего пути. Приведено описание алгоритма поиска и структур данных.*

*Ключевые слова:* программная реализация алгоритма поиска кратчайшего пути, разработка, граф, алгоритм Дейкстры, общественный транспорт.

### Abstract

*This article describes software realization and implementation for the problem of finding the most efficient route between two different points "A" to point "B" using public transport system. The developed program is intended to be used inside complex information system, informing about public transport routes. Using set lists of start and end points (the terminals installed at bus stops and connected to the information system are used as start points; stopping complexes or sights of the city are used as end points) and public transport motion data, the program identifies and analyzes the route between all pairs of start and end points. The best routes are displayed on the terminal of bus stops at the request of residents and guests of the city by means of an integrated information system. The algorithm and the solution to a problem of finding the shortest way are considered in this article. The description of the search algorithm and data structures are provided.*

*Keywords:* software implementation of the algorithm for finding the shortest path, development, graph, Dijkstra's algorithm, public transport.

Время – это наш самый ценный и дефицитный ресурс. Перед каждым человеком регулярно возникает задача оптимизации его использования. От того, как конкретный индивид справляется с этой задачей, зависит его успешность в постоянно меняющемся окружающем мире.

В условиях высокой загрузки основных городских транспортных магистралей жители мегаполисов тратят много времени на то, чтобы добраться из одной точки города в другую. В случае если это не привычный маршрут, необходимо потратить время на определение нужного вида транспорта, поиск близлежащего остановочного комплекса, выбор нужного маршрута и направления движения. И такие вопросы возникают не только у гостей, не знакомых с маршрутной сетью городского общественного транспорта, но даже у коренных жителей, попавших в незнакомый район.

Не стоит также забывать и о внешних факторах. Это и загруженность на дорогах, которая зависит от времени суток, и пропускная способность дорог, и дорожно-транспортные происшествия, различные перекрытия дорог, изменения маршрутной сети (например, в связи с плановыми и внеплановыми дорожными работами), сход транспортных средств с маршрута и т.д.

Информирование об оптимальных маршрутах между различными точками позволяет помочь пользователям общественного транспорта оптимизировать свои временные затраты на планирование перемещения по городу.

Для решения этой задачи могут быть использованы информационные терминалы в составе остановочных комплексов общественного транспорта.

Целью настоящей работы является разработка прототипа программы поиска оптимального маршрута из точки «А» в точку «Б» для всех пар заданных начальных и конечных точек. Полученные в соответствии с разработанным алгоритмом сведения будут визуализироваться встроенными аудио-визуальными средствами информационного терминала.

Решением задачи является программная реализация алгоритма поиска кратчайшего пути между любыми двумя пунктами назначения (вершинами графа). Для своей работы программа использует массив табличных данных, хранящийся в базе данных. В начале работы программы данные таблиц считываются в оперативную память, после чего для каждой пары начальных и конечных точек производится поиск кратчайшего пути. Результатом работы программы для каждой пары начальных и конечных точек является запись в таблице с результирующими значениями: виды транспорта, номера маршрутов, пункты пересадок, ожидаемое время в пути между этими точками. Итоговая таблица с результирующими значениями записывается в базу данных. С заданной периодичностью средствами информационной системы соответствующая часть таблицы с результирующими значениями передается на информационные терминалы, установленные на остановочных комплексах для последующего отображения по запросу пользователей.

Для решения задачи в базе данных предварительно формируется массив исходных данных – начальные и конечные остановки, виды транспорта, маршруты транспорта и т.д. Объектом, для которого производится выбор оптимальных маршрутов в данной работе, является город Екатеринбург. На карту города накладывается координатная сетка, узлы которой становятся вершинами графа, в котором осуществляется поиск. Каждая остановка соотносится какой-либо одной вершине графа. Ребра графа поиска (перемещения общественного транспорта между остановками) формируются по данным, поступающим из РНИС (региональная навигационная информационная система), в которую попадают от датчиков системы ГЛОНАСС, установленных на общественном транспорте. Для каждого перемещения между вершинами можно зафиксировать вес (время перемещения по данным системы ГЛОНАСС). Таким образом, можно рассматривать данный набор сведений как ориентированный взвешенный граф.

Для поиска решения оптимального маршрута между вершинами графа могут применяться различные алгоритмы. Выбор алгоритма зависит от характеристик графа. Поскольку в данной задаче рассматривается ориентированный взвешенный граф, целесообразно использовать классический алгоритм Дейкстры. Алгоритм позволяет находить кратчайшие пути от одной из вершин графа до всех остальных. Алгоритм широко применяется в программировании и протоколах маршрутизации.

Описание алгоритма:

Каждой вершине из множества точек графа поиска - сопоставляется метка – минимальное известное расстояние от этой вершины до «А» (начальная остановка). Вершина «А» имеет метку «0», так как из неё осуществляется поиск. Метки остальных вершин на момент начала работы алгоритма равны бесконечности, так как расстояния до них неизвестны. Все вершины графа отмечаются как «не посещённые». Алгоритм работает пошагово – на каждом шаге он «посещает» одну вершину и пытается уменьшить её метку. Если все вершины графа поиска посещены, алгоритм завершает свою работу. В противном случае, из еще не посе-

щенных вершин выбирается вершина 'х', имеющая минимальную метку. Рассматриваем все возможные маршруты, в которых 'х' является предпоследним пунктом. Вершины, в которые ведут ребра из 'х', назовем соседями этой вершины. Для каждого соседа вершины 'х', кроме тех, которые отмечены как «посещенные», рассмотрим новую длину пути, равную сумме значений текущей метки 'х' и длины ребра, соединяющего 'х' с этим соседом. Если полученное значение длины меньше значения метки соседа, заменяем значение метки на полученное значение длины. Рассмотрев всех соседей вершины 'х', пометим ее, как «посещенную» и повторим шаг алгоритма.

В рамках работы разработан и запрограммирован алгоритм решения данной проблемы, создана база данных на MS SQL server.

Для учета изменения оптимального маршрута с учетом типовой загруженности городских улиц алгоритм выполняется девять раз по данным из РНИС для различных интервалов времени одного дня. В этом случае вычисленные данные будут объективными и будут отражать ситуацию на дорогах города для различных интервалов времени в течение дня. Прототип программы выполняет предварительные действия и непосредственно вычисления.

#### Предварительные действия:

1) Заполняем данными таблицы с исходными данными (справочники) и вносим справочники в базу данных (справочник остановок, справочник маршрутов, справочник исходных точек, справочник конечных точек).

2) Создаем массив, состоящий из опорной и граничных точек, записываем его в базу данных.

3) Создаем таблицу координатной сетки, записываем её в базу данных.

4) Создаем и заполняем таблицу исходных (начальных) точек, записываем её в базу данных (\*если у исходной точки несколько остановок, мы изначально каждой присваиваем вес «1»).

5) Создаем и заполняем таблицу остановок, записываем её в базу данных.

6) Создаем и заполняем таблицу отрезков, записываем её в базу данных.

7) Создаем и заполняем таблицу конечных точек, записываем её в базу данных (\*в векторе обойденных точек берем каждое находим вес у каждой ID ост ближайшей)

8) Создаем и заполняем справочник маршрутов, записываем его в базу данных.

#### Блок вычислений

1. Считываем данные из справочников.

2. Дублируем справочник остановок, сделав выборку по «ID точки координатной сетки», для получения всех остановок, которые относятся к данной точке.

3. Получаем данные из РНИС (находятся в массиве данных):

- Импортируем данные в СУБД;
- Удаляем избыточные (храним данные только за последние 8 дней)
- Анализируем каждый маршрут – за какое время проходит общественный транспорт между двумя остановками.

4. Строим координатную сетку для каждого маршрута (поля: ID точки координатной сетки, цена пересадки, вектор пересадочных точек, вес перехода в следующую точку – анализ данных из РНИС).

5. Строим вектор обойденных точек.

6. Строим вектор рассматриваемых точек (текущих).

7. Отправляем результат в СУБД.

Структура базы данных, используемая программой, состоит из справочника остановок, справочника маршрутов, справочник исходных точек, справочника конечных точек. таблицы «Транспорт», таблицы «Признак остановки», таблицы «Остановки маршрутов», таблицы «Отрезки» и таблицы «Информационный терминал».

В таблице 1 отображены сущности таблиц, их поля с типами данных.

На рис. 1 представлена архитектура комплексной информационной системы информирования о маршрутах общественного транспорта города Екатеринбурга. Программный мо-

дуль связан с СУБД и производит периодический (1 раз в сутки) пересчет оптимальных маршрутов между начальными и конечными точками для различных интервалов времени суток. Вычисленные маршруты хранятся в отдельной таблице, данные из которой с заданной периодичностью средствами информационной системы передаются на информационные терминалы. Доступ к данным указанной таблицы также доступен для специализированных веб-сервисов информационной системы, позволяющих пользователям мобильных и настольных компьютерных систем получить информацию о кратчайших маршрутах в режиме реального времени.

Таблица 1

Структура таблиц базы данных

Сущность	Атрибут	Тип данных
Маршруты	id_маршрута	int
	Номер маршрута	text
	id транспорта	int
Остановки	id остановки	int
	Название остановки	text
	Долгота	float
	Широта	float
	Описание	text
	id коорд сетки	int
Остановки маршрутов	id маршрута	int
	id остановки	int
	Номер по порядку	int
Транспорт	id транспорта	int
	Наименование	text
Признак остановки	id признак_ост	int
	признак	bit
	описание	text
Исходные точки	id исх точ	int
	Наименование	text
	Долгота	float
	Широта	float
	id ближ_ост	int
Конечные точки	id кон точ	int
	Наименование	ext
	Долгота	float
	Широта	float
	Описание	text
	id признак_ост	int
	id ближ_ост	int
Отрезки	id_ост1	int
	id_ост2	int
	Время прохождения	tinyint
Информационный терминал	id_инфотерминал	int
	Долгота	float
	Широта	float

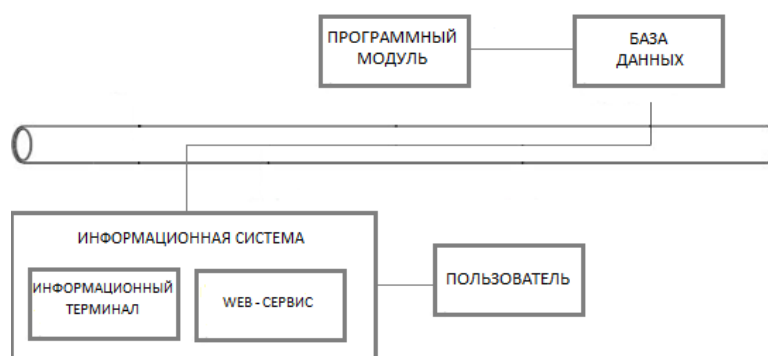


Рис. 1. Архитектура информационной системы

Разработанный прототип программы поиска оптимального маршрута из точки «А» в точку «Б» показал возможность практической реализации извлечения и анализа «сырых» данных из базы данных региональной навигационной информационной системы с последующим формированием структур данных, позволяющих построить оптимальные маршруты перемещения на общественном транспорте между любыми двумя точками в пределах маршрутно-транспортной сети города Екатеринбурга.

Данный прототип также может быть адаптирован для использования в других городах не только Российской Федерации, но и за рубежом.

#### Список использованных источников

1. Электронный учебник [сайт]. URL: [http://www.math.mrsu.ru/text/courses/method/osn\\_pon\\_teor\\_graph.htm](http://www.math.mrsu.ru/text/courses/method/osn_pon_teor_graph.htm)
2. Электронный учебник [сайт]. URL: <http://www.isa.ru/transnet/TrafficReview.pdf>
3. Электронный учебник [сайт]. URL: <http://kvodo.ru/dijkstra-algorithm.html>
4. Электронный ресурс [сайт]. URL: <http://edulib.pgta.ru/>
5. Электронный ресурс [сайт]. URL: <https://habrahabr.ru/>

УДК 681.518.5

**А. Н. Яранцев**

ФГАОУ ВО «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина», г. Екатеринбург, Россия

#### РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ЛАБОРАТОРНОЙ УСТАНОВКИ ПО ИЗУЧЕНИЮ БЕСКОНТАКТНОГО ИЗМЕРЕНИЯ ТЕМПЕРАТУРЫ

##### Аннотация

Лабораторная установка предназначена для изучения бесконтактного измерения температуры. В процессе работы выполняется считывание сигналов, их обработка, хранение и представление пользователю в удобном виде. Первичными являются электрические сигналы с пирометра и термоэлектрического преобразователя, которые обрабатываются контроллером и рабочей станцией при помощи разработанного программного обеспечения.

Ключевые слова: программируемый логический контроллер, ПЛК, рабочая станция, WinCC.

##### Abstract

The laboratory stand is designed to study non-contact temperature measurement. In the process involves reading the signals, their processing, storage and presentation to the user in a con-